Print Request Result(s)

Printer Name: ran_3c18_gbrcptr
Printer Location: ran__3c18
Number of Copies Printed : 1

- US20040107272: Ok
- US20020184349: Ok
- US20020055967: Ok

OK Back to List

709/221
202
705/51

Freeform Search

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: L15 and (contact near information)

Display: 50 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ **Hit List** ● **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Thursday, January 20, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L17	L16 and susbcrib$	0	L17
L16	L15 and (contact near information)	14	L16
L15	(status near system) and (status near server)	166	L15
L14	((20020052921).pn.) and (contact near information)	2	L14
L13	L12 and (contact near information)	1	L13
L12	(20020064149).pn.	2	L12
L11	L9 and availab$	22	L11
L10	L9 and availab$	22	L10
L9	l2 and contact	22	L9
L8	L6 and availab$	1	L8
L7	L6 and status	0	L7
L6	L3 and contact	1	L6
L5	L3 and (dynamic near contact)	0	L5
L4	L3 and meeting	0	L4
L3	L2 and diacakis	1	L3
L2	L1 and subscriber	22	L2
L1	(contact near information) and (available near management)	49	L1

END OF SEARCH HISTORY

First Hit Previous Doc Next Doc Go to Doc#

□ Generate Collection Print

L14: Entry 1 of 2 File: PGPB May 2, 2002

DOCUMENT-IDENTIFIER: US 20020052921 A1
TITLE: Systems and methods for managing contact information

Abstract Paragraph:

A method and system for secure management of contact information is disclosed. A service provider maintains a database with subscribers' updated contact information, such as address book entries and other personal and business information. When a user's information changes, subscribers on a recipient list are automatically updated without any user intervention. The subscribers' anonymity is assured by correlating contact information in the provider's database with encrypted subscriber ID's, such as hashed e-mail addresses.

Pre-Grant Publication (PGPub) Document Number:
20020052921

Summary of Invention Paragraph:
[0002] The present invention relates generally to systems and methods for managing personal information, and more particularly to systems and methods that controllable and securely acquire, handle and maintain contact information. The proposed contact information management systems and methods operate with e-mail applications, personal information managers and wireless devices.

Summary of Invention Paragraph:
[0008] According to another aspect of the invention, in a method for secure management of contact information via a network, a client prepares an e-mail with a recipient's e-mail address, and indicates in the e-mail that contact information is included. A server in communication with the client vi the network associates a transaction ID with the contact information, stores a hash of the recipient's e-mail address received from the client and associates the hashed e-mail address with the transaction ID. The server then forwards the e-mail with the transaction ID to the recipient, receives from the recipient a message that includes the transaction ID, produces a hash of the recipient's e-mail address received from the recipient, and if the hashed e-mail address and transaction ID received from the recipient match the stored hash of the recipient's and the transaction ID, forwards the contact information to the recipient.

Summary of Invention Paragraph:
[0009] According to yet another method for secure management of contact information at a server connected to a network, the server receives contact information from a client via the network, associates a transaction ID with the received contact information, computes at least one hashed e-mail address of an intended recipient specified by the client, and associates the hashed e-mail address with the transaction ID. The server may then compute a hash of the recipient's e-mail address received from the recipient, match the hashed recipient's e-mail address received from the recipient with the transaction ID, and transmit the contact information to the recipient.

Summary of Invention Paragraph:
[0010] According to yet another aspect of the invention, a system for disseminating contact information via a network includes a server that is connected to the network and stores the contact information. A sender's computer is also connected to the network and transmits via the network to a recipient's e-mail address an e-mail that includes an identifier associated with the sender's contact information. A recipient's computer that is connected to the network receives the e-mail with the identifier and accesses the server via the network and retrieves from the server the contact information by providing the server with the identifier. The server can store an association between the contact information and a recipient's e-mail address based

on an encrypted value of the recipient's e-mail address, such as a hash of the e-mail address. In this way, the server does not know the identity of the recipient before the recipient requests the contact information from the server.

Summary of Invention Paragraph:
[0011] Embodiments of the invention may include one or more of the following features. The sender may indicate by a message appended to the e-mail and/or by a field checked in the e-mail window that contact information is included. The contact information may be targeted for a specified recipient or for a plurality of recipients listed on a recipient list. The server may retain copies of the contact information and/or a history of the contact information, with the history being derived from transaction ID's. The recipient --in response to receiving the contact information from a sender--may transmit updated contact information to the sender. The client may specify different security settings for the contact information, with respective contact information displayed at the recipient depending on the security setting. The server, when receiving updated contact information for a particular recipient on a recipient list, may automatically send an e-mail to the remaining recipients on the list with the updated contact information. The server may also produce a hash of other recipient's e-mail addresses that are linked to the recipient's e-mail address and match the hash of the linked e-mail addresses with the transaction ID, with the server forwarding the contact information to the other e-mail addresses of the recipient.

Brief Description of Drawings Paragraph:
[0014] FIG. 1A depicts a system for managing contact information via a network;

Brief Description of Drawings Paragraph:
[0016] FIG. 2 is a high-level flow diagram for managing contact information;

Brief Description of Drawings Paragraph:
[0017] FIG. 3 is a detailed flow diagram for a registered client sending contact information;

Brief Description of Drawings Paragraph:
[0018] FIG. 4 is a detailed flow diagram for a registered client receiving contact information;

Brief Description of Drawings Paragraph:
[0019] FIG. 5 is a detailed flow diagram for an unregistered client receiving contact information;

Brief Description of Drawings Paragraph:
[0020] FIG. 6 shows the data flow between server and client for exchanging contact information;

Brief Description of Drawings Paragraph:
[0021] FIG. 7 shows the data flow between server and client for updating contact information;

Brief Description of Drawings Paragraph:
[0022] FIG. 8 depicts an exemplary contact information in form of a "livecard" sent to a subscriber with an installed e-mail client add-in;

Brief Description of Drawings Paragraph:
[0023] FIG. 9 depicts an exemplary contact information in form of a "livecard" sent to a subscriber without an e-mail client add-in; and

Brief Description of Drawings Paragraph:
[0024] FIG. 10 depicts an exemplary contact information in form of a "livecard" received by a subscriber without an e-mail client add-in.

Detail Description Paragraph:
[0025] The invention is directed to methods and systems for managing contact information. In particular, the methods and systems for managing contact information described herein can securely acquire, maintain and update contact information using e-mail protocols.

Detail Description Paragraph:

[0026] Referring first to FIG. 1A, a system 10 includes client machines 12, 13 with associated local databases 11, 14, wherein the local databases 11, 14 may be internal to the client machines 12, 13. The system also includes a server 17, such as a server of a trusted party administering the contact information. The contact information can include, for example, personal, biographical or business-related data of the type commonly found on printed business cards, but may contain any other information, such as image, audio and video data, that may be of interest to a selected third party. The server 17 and the client machines 12, 13 can communicate with one another over a network 15, such as the Internet or a LAN. The server 17 connects to a proprietary database 16 which securely stores user identities and user contact information, as will be described in detail below.

Detail Description Paragraph:
[0030] In the following, it will be assumed that a user connects to a contact information service provided by the server 17. The user will subsequently also be referred to as subscriber or sender. The client machines host client application programs that can include a so-called "plug-in" 20 capable of analyzing transmitted and received e-mail messages to detect the presence of appended contact information. The plug-in can also separate the appended contact information from the e-mail body text so as to communicate the contact information to the server, while displaying only the e-mail body text to the subscriber. The operation of the system with and without a plug-in will be described below.

Detail Description Paragraph:
[0031] As depicted in FIG. 1B, the client machines 12, 13 may host an e-mail client add-in 20, also referred to as "plug-in", in form of a DLL that provides one or more particular functions, such as interfacing the e-mail contact management tasks with standard application programs, such as Microsoft Outlook.TM. or the Palm.TM. Desktop. The plug-in 20 interfaces with the network 15 for communication with server 17 through an HTTP Manager 26. The HTTP Manager 26 exchanges messages, such as e-mail and web page content, with a Data Manager 28 that interfaces with the local database 11 via an application program interface (API) to control access to the database 11. Any communication, such as e-mail with appended contact information, received by the Data Manager 28 can be transmitted via message queue 32 to a GUI Manager 22 adapted to provide, for example, a number of dialog boxes, information displays and interfaces for facilitating the convenient viewing, accessing, managing, forwarding and synchronizing the contact information.

Detail Description Paragraph:
[0032] A user can use the command available on the GUI Manager 28 to instruct an Event Manager 24 to schedule certain tasks relating to the management of contact information, such as having the Data Manager 28 synchronize contact information with the server 17 and/or a client machine.

Detail Description Paragraph:
[0033] Referring now also to FIG. 2, the following scenarios can be encountered in the system 10: (1) the subscriber has an installed plug-in and sends and receives contact information; (2) the subscriber does not have an installed plug-in and sends and receives contact information; and (3) the recipient of the e-mail with the appended contact information is not a subscriber of the services provided by the server 17. It will be assumed that only a valid subscriber can send the contact information using the server 17.

Detail Description Paragraph:
[0034] Referring now back to FIG. 1A, an exemplary transmission of contact information from a sender's (subscriber) client machine 12 to the client machine 13 of a recipient who may or may not be a subscriber, is indicated by the broken arrows 1, . . . , 6. In the first transmission corresponding to arrow 1, it is assumed that the sender 12 has activated a SEND button, for example, in a browser window, indicating that he wishes to send contact information, such as an electronic business card, to the recipient 13. For security reasons, the database 16 of server 17 stores a correlation of the contact information of the sender and a hash and not the plain e-mail address(es) of the e-mail address(es) of the sender which are provided by the plug-in 20 installed in client machine 12.

Detail Description Paragraph:
[0035] In a first communication between the sender 12 and the server 17 via network 15, as indicated by arrow 1, the sender 12 sends the e-mail address(es) of the recipient(s), for

example, in form of a hash generated by the plug-in of client machine 12, to the server 17. The server 17 searches database 16 for the contact information associated with the recipient(s) based on the transmitted e-mail address(es) of the recipient(s), as indicated by the arrow 2. This arrangement shields the identity of the receiving parties of the contact information from the service operating the server 17 until the contact information transmitted from the sender 12 to the recipient 13 is accepted by the recipient 13. It will be understood that the system 10 can include a plurality of senders 12 and recipients 13. The server 17 then responds with information relating to the contact information that is to be appended to the e-mail, as will be described in more detail below with reference to FIG. 3.

Detail Description Paragraph:
[0036] Optionally, the communication steps 1 to 3 above can be omitted, with the interaction and exchange of contact information between the sender 12 and the recipient 13 initiated in step 4, as described below.

Detail Description Paragraph:
[0037] Having composed the e-mail text message and appended the contact information, the subscriber 12 sends the e-mail to the recipient 13, as indicated by arrow 4. This communication can take place via an unsecured communication channel.

Detail Description Paragraph:
[0038] If the recipient is a valid subscriber of the service provided by the server 17, then the recipient 13 is made aware of receipt of new contact information by, for example, an e-mail message and/or a message displayed in a browser window, which occurrence will be communicated to the service provider, server 17, as indicated by arrow 5. The service provider 17 will respond back that new contact information is available and will transmit the new contact information to the recipient, for example, for display in a browser window, as indicated by arrow 6. The recipient can accept or decline the transmitted contact information.

Detail Description Paragraph:
[0039] If the recipient is not a valid subscriber of the service provided by the server 17, then the recipient 13 is made aware of receipt of new contact information by, for example, an URL link appended to the e-mail body and including the service provider's e-mail address. By clicking on this link, the non-subscriber accesses the service provider 17, as indicated by arrow 5. As will be explained below, the e-mail address includes an extension with a checksum generated by the sender and including the recipient's e-mail address or a hash of the recipient's e-mail address. The service provider can also compute a hash of the recipient's e-mail address and forwards the contact information to the recipient only if the two checksums or hash values agree, arrow 6.

Detail Description Paragraph:
[0040] Referring now to FIG. 3, a process flow 300 for sending e-mail with appended contact information is illustrated. The process flow 300 also illustrates the operation of the plug-in 20. A subscriber (sender) composes an e-mail message which may or may not include contact information, step 310. The e-mail message is processed by the senders client which detects if an e-mail message is appended, steps 312 and 314, respectively. If no contact information is appended, step 316, then the e-mail message is sent to a recipient, step 330. On the other hand, if the presence of appended contact information is detected in step 314, then the contact information is identified, step 318. In the event that the sender's client does not include a plug-in, as determined in step 320, the sender is prompted to manually insert the link text for the contact information and append the link text to the e-mail, step 322. As seen from FIG. 8 and briefly described above with reference to FIG. 1, the link text can include the e-mail address of the service provider and an extension identifying the recipient. It should be pointed out that sending the contact information without the plug-in weakens the security of the transmission.

Detail Description Paragraph:
[0041] Conversely, if the sender's client has a plug-in, then the link text will be inserted by the plug-in, step 324, together with a check sum computed in step 326 from the ID associated with the sender's contact information, a sender's secret ID and the recipient's e-mail address. This checksum will be used later to authorize the intended recipient to receive the contact information. An optional header can be inserted in the e-mail to indicate the presence of contact information, step 328, before sending the e-mail, step 330.

Detail Description Paragraph:
[0042] Referring now to FIG. 4, a recipient who is a valid subscriber, receives an e-mail with appended contact information, step 410. If no contact information is appended (this is not of interest for the present application), then the recipient just reads the e-mail, but does not communicate further with the server. The process 400 branches at step 412, depending whether or not a plug-in is provided at the recipient's client.

Detail Description Paragraph:
[0043] If the recipient has a plug-in, then the recipients client searches for an identifier in the e-mail, indicating that contact information is appended to the e-mail, step 414. The plug-in extracts the transfer string of the contact information, step 416, and strips off the extraneous text in the e-mail body not pertaining to the contact information, step 418. The the recipients can then process the contact information by (1) accepting or declining to the contact information; (2) instructing the server to update to the contact information in the recipients account; and/or to update the server to include a the contact information and link the contact information with the recipients (hashed) account identifier, step 420. The recipient is able to read the text body of the e-mail message, step 422, optionally with a note indicating to the reader that updated contact information was appended to the e-mail message.

Detail Description Paragraph:
[0044] If the recipient does not have a plug-in, as determined and step 412, then the recipient will notice the presence of links appended to the text of the e-mail message which include, for example, the Web address of the server that received the updated contact information from the sender, step 424. The recipient then clicks on the link associated with the recipients e-mail address, as displayed in the link, step 426. In response, the server can send a message to the recipient, according to recipient to the existence of updated contact information, which the recipient can either accept or decline, step 428. If the recipient accepts the contact information, then the server will associate the sender's contact information with the recipients account as a provisional contact, with the contact information being officially added at the next login of the recipient, step 430. Because of the absence of the plug-in, the link appended to the e-mail body text cannot be stripped from the message.

Detail Description Paragraph:
[0045] Referring now to FIG. 5, a process of 500 is described wherein contact information is sent to a recipient who is not a subscriber, step 510. However, although the recipient is not a subscriber, the recipient may have a plug-in, for example, left on the recipients client machine from a previous subscriber or subscription. If a plug-in is present, as determined in step 512, then the plug-in checks the validity of the ID's and verifies the checksum, step 524. If, for example, the checksum does not match the recipient's ID, then the sender may receive a message indicating a problem with the intended recipient, step 526. If the ID's and the checksum match, then the plug-in extracts the recipients e-mail address, step 528. If the e-mail address is stored in the server, as tested in step 530, then the contact information is added to the recipients account, step 532. If, on the other hand, the recipients e-mail address is not stored in the server, then the server sends a message to the recipient, step 518, informing the recipient of the presence of updated contact information, which the recipient can either accept or decline, step 520. If the recipient accepts the contact information, then the server will e-mail the recipient a link to the contact information and set up a provisional account, optionally inviting the recipient to subscribe to the services provided by the server.

Detail Description Paragraph:
[0046] If the recipient is not a subscriber and does not have a plug-in, then the the recipient will notice the presence of links appended to the text of the e-mail message which include, for example, the Web address of the server that received the updated contact information from the sender, step 514. The recipient then clicks on the link associated with the recipients e-mail address, as displayed in the link, step 516. In response, the server displays a message, step 518, informing the recipient of the presence of updated contact information, which the recipient can either accept or decline, step 520. If the recipient accepts the contact information, then the server will e-mail the recipient a link to the contact information and set up a provisional account, optionally inviting the recipient to subscribe to the services provided by the server.

Detail Description Paragraph:
[0047] FIGS. 6 and 7 depict the flow of information between the user/e-mail client and the server/server database when an e-mail client receives e-mail which may include contact information (FIG. 6), and when a user elects to update contact information in the server database and communicate the updates to users included in a list relating to this contact information (FIG. 7), respectively.

Detail Description Paragraph:
[0048] Referring now to FIG. 6, in a process 600, an e-mail client receives e-mail, step 610, and processes the receives e-mail, step 612. If no contact information is included in the e-mail, as determined in step 614, then only the e-mail is displayed without any further action on part of the e-mail client, step 620. If the e-mail, the other hand, includes contact information, as determined in step 614, then the server is queried to determine if the user already has the contact information, steps 616 and 618. As before, no action on part of the e-mail client is required if the user already has the contact information. On the other hand, if the user does not already have the contact information, then details of the contact information are displayed by the e-mail client, step 622, and the user can either accept or decline the contact information, step 624. If the user declines, step 626, then the process may notify the server, step 630, with the action being recorded in the server database, step 632, and the e-mail attachment with the contact information can be deleted from the e-mail body, step 624. Conversely, if the user accepts the (updated) contact information in step 624, the information is saved by the e-mail client, step 628, the server is notified, step 630, with the action being recorded in the server database, step 632, as before. The e-mail attachment can also be deleted, step 634.

Detail Description Paragraph:
[0049] The contact information stored in the local database and the server database can be edited and/or updated in several ways. In the exemplary process 700 depicted in FIG. 7, a user selects an address book entry, step 710, either manually or from a list of address book entries stored in the e-mail client, step 712, and edits the entry, step 714. The edits are saved in the local database of the e-mail client, step 716, and the edited contact information is transmitted to the server for synchronization with the server database, step 718. The server temporarily stores the edited contact information, step 720, and retrieves a list of recipients, which is preferably a hash of the recipients' e-mail addresses, that are associated with this contact information and need to be updated, step 722. The list of recipients may be retrieved from the server database, step 724. The list of the recipients to be updated can then be displayed to the editing user, step 726, and for the edited by the user, step 728. After the editors complete, the e-mail client notifies the server of the actions taken, step 730, but after the server updates the list of the recipients based on the edits, step 732. The updated contact information for the recipients is stored on the server database, step 734.

Detail Description Paragraph:
[0050] FIGS. 8-10 depict various exemplary formats for textual contact information sent via e-mail to a recipient. In the embodiment illustrated in FIG. 8, the contact information is sent in form of a so-called "livecard" to a recipient who is a subscriber and equipped with a plug-in. The plug-in recognizes the last text entry "My business livecard is included" and will obtain from the server the updated contact information of the sender. Conversely, as depicted in FIG. 9, if the recipient is a subscriber, but the sender does not have a plug-in, then the sender includes the service provider's e-mail address with an extension showing the sender's ID, The recipient can obtain the updated contact information by clicking on the link.

Detail Description Paragraph:
[0051] In another embodiment, a recipient may not have a plug-in, as shown in FIG. 10. The recipient then clicks on his/her e-mail address link to obtain the contact information update from the service provider.

Detail Description Paragraph:
[0052] While the invention has been disclosed in connection with the preferred embodiments shown and described in detail above, various modifications and improvements thereon will become readily apparent to those skilled in the art. For example, contact information ("livecards") can be categorized by populating certain fields, i.e., "livecards" can be associated with different functions and purposes. One set of livecards can be for private use, whereas another set of livecards can be intended for business use, optionally with different logos. The user

can compile a master list of personal details select subsets thereof for various types of cards, each of which carry the selection of these fields. Before a field in a card which carries that field that is to be updated, is actually updated, the user can view the new update as well as the name of the person(s) receiving the update before sending it out. The user has the chance to take a person off the update list and can send an update message out with the update, as shown in FIGS. 8-10.

Detail Description Paragraph:
[0053] The user also has the opportunity to permanently remove a person from the contact information list. In the proposed embodiments, contact entries with e-mail addresses are being processed, i.e., the present process works with e-mail folders. A default setting could be that the contact information is being forwarded, with the user being able to specify which contact information is being sent. Updates can be time-triggered with configurable settings or performed manually. Each e-mail that is received is being scanned for a livecard tag, without interrupting the user's activities. The user should also be able to read the e-mail body text off-line without being interrupted by the life livecard utility.

Detail Description Paragraph:
[0054] Livecards have associated therewith certain security levels. For example, a lowest security level may be where all information is displayed via URL code and anyone can subscribe to it; a higher security level may display the contact information via URL or code, anyone can request subscription, but only intended recipients can subscribe to; and at a still higher security level, nothing is displayed via URL or code, but subscription is available for intended recipients. To enforce the security code, every livecard can carry a public code that is printed on the livecard and activated when the recipient activates the livecard URL link. The service provider's web site will give access to the card according to the security level set for that card. For example, the code "876" displayed in FIG. 9 after the name "Bob" can represent such a security code.

CLAIMS:

1. Method for secure management of contact information via a network, comprising: attaching contact information to an e-mail having a recipient's e-mail address, associating a transaction ID with the contact information, associating a first hash of the recipient's e-mail address with the transaction ID, transmitting the e-mail with the transaction ID to the recipient via the network, producing a second hash of the recipient's e-mail address, and if the second hash of the recipient's e-mail address and transmitted transaction ID match the first hash of the recipient's e-mail address and the transaction ID, forwarding the contact information to the recipient.

2. The method of claim 1, wherein attaching the contact information includes checking if the recipient already has a most recent copy of the contact information.

3. The method of claim 1, wherein a presence of the contact information is identified in the e-mail by a message appended to the e-mail.

4. The method of claim 1, wherein a presence of the contact information is identified in the e-mail by a field checked in an e-mail window.

5. The method of claim 1, wherein the contact information is directed to a specified recipient.

6. The method of claim 1, wherein the contact information is directed to a plurality of recipients listed on a recipient list.

7. The method of claim 1, further including retaining a history of the forwarded contact information.

8. The method of claim 7, wherein the history of the contact information is associated with the transaction ID.

9. The method of claim 1, wherein the recipient scans a received e-mail for an indication of the presence of updated contact information.

10. The method of claim 1, wherein a new contact information entry is created at the recipient if the recipient does not currently have the contact information.

11. The method of claim 1, wherein the recipient--in response to receiving the forwarded contact information--transmits updated recipient contact information.

12. The method of claim 1, further including associating security settings with the contact information, with the security settings determining the forwarded contact information displayed at the recipient.

13. The method of claim 6, wherein when a recipient on the recipient list receives updated contact information, said updated contact information is automatically sent to the remaining recipients on the recipient list.

14. The method of claim 1, wherein the recipient has additional e-mail addresses that are linked to the recipient's e-mail address, and the updated contact information is forwarded to the linked e-mail addresses using the same transaction ID.

15. Method for secure management of contact information over a network, comprising: associating a transaction ID with the contact information, computing a hash of an e-mail address of an intended recipient of the contact information, and associating the hashed e-mail address with the transaction ID.

16. The method of claim 15, further comprising: when the intended recipient requests the contact information, computing a second hash of the recipient's e-mail address associated with the request, matching the second hashed recipient's e-mail address with the first hashed e-mail address for the transaction ID, and if the first and second hashed e-mail addresses agree, transmitting the contact information to the recipient.

17. A system for disseminating contact information via a network, comprising: a server connected to the network and storing the contact information, a sender's computer connected to the network and transmitting via the network to a recipient's e-mail address an e-mail that includes an identifier associated with the sender's contact information, and a recipient's computer connected to the network and receiving the e-mail with the identifier, the recipient's computer accessing the server via the network and retrieving from the server the contact information by providing the server with the identifier.

18. The system of claim 17, wherein the identifier and the recipient's e-mail address define the contact information retrieved from the server.

19. The system of claim 17, wherein the server stores an association between the contact information and a recipient's e-mail address based on an encrypted value of the recipient's e-mail address.

20. The system of claim 17, wherein the identifier is a transaction ID of the contact information.

Previous Doc Next Doc Go to Doc#

L16: Entry 3 of 14 File: PGPB Jun 3, 2004

DOCUMENT-IDENTIFIER: US 20040107272 A1
TITLE: Method and system for automatically configuring a client-server network

Detail Description Paragraph:
[0082] In order to handle multiple communications, it is contemplated that server 44 may be distinguished from other servers of group 40 by a server identification number (ServerID). The ServerID may be any set of alphanumeric data, converted into machine language that identifies and distinguishes one server from the other based upon attributes, such as the location, capacity, services, clients assigned, system status and other criteria. The ServerID acts as a license plate that is used by the system 10 to identify each server that is used as part of the server group 40 so that any inbound communication will be directed to the proper server. The ServerID for each server of the group 40 is stored in a server list maintained by the database group 42. Other means for distinguishing the identity of one server of the group 40 from another may be used.

Detail Description Paragraph:
[0090] A central storage means 86, such as a database file or "brain" is located in database server 76 to releasably retain account data 88. The account data 88 comprises information associated with the services made available to the client 14, such as the account settings 66 and client account information 89. The account information 89 comprises attributes of a client's account, such as the name of the particular user or client, account verification and authentication information 90, billing information 92, a list of the settings 96 of the services desired by the client 14, and the like. The authentication information 90 includes credentials that will uniquely identify the client 14 to the system 10 and grant access to the network 16 when the client 14 logs in. The authentication information 90 may comprise the end-user's name, passwords, email address, domain names, contact information, and similar data so that the system 10 can distinguish one client from another. The billing information 92 includes payment information such as the payment options (i.e., credit card, checking account), credit card number, card expiration date, name and address of cardholder, and other methods in which the client can be charged for services provided by the system 10.

Detail Description Paragraph:
[0113] The web page is preferably, but not necessarily, maintained on the controller 108. The web page has an "Order Form" that is generated by an account setup program running on one of the servers of group 40. The order form is used so that the client 14 may enter information used for the account settings 66, account data 88, account information 89, and other information necessary in setting-up the account. For example, the order form will solicit from the client 14 contact information, billing information, and information such as the services that the client 14 is requesting. The contact information will include the username(s), password(s) requested, email address(s) requested, email password(s) requested, domain name, present email address, the company name, first name, last name, address, city, state province, etc. The billing information which may include the credit card company, account number, card expiration date and the name and address of the card holder. The billing information will travel through a Secure Socket Layer (SSL) enabled web page which allows the client to submit sensitive information. The order form will include "slots" or areas on the web page in which all of the above information may be entered or typed in by the client 14.

Detail Description Paragraph:
[0115] Next, the client 14 uses the pointer (mouse) 28 to "point and click" on the submit button that is displayed on a portion of the order form. When the client 14 clicks the submit button, the data that was entered by the client 14 is transmitted utilizing the SSL protocol to the controller 108 for verification. During the verification process, the authentication or verification program running on the SSL server will review the contact information and other

data that was entered by the client 14 to ensure that all pertinent information is entered correctly. For example, if the client 14 omitted a name or did not include the credit card number, the verification program will generate an "Error Message", at block 115. The Error Message is generated by the authentication program and transmitted to the client 14. The client 14 receives the Error Message which is displayed on the monitor 30 so that the client 14 may correct the error. Preferably, the Error Message will display specifically what the problem is to the client 14, such as which specific information was omitted.

Detail Description Paragraph:
[0116] Once all of the necessary information is verified at block 116, the SSL server will perform security checks based on the information provided. The security check is performed by a security verification program running on the server at block 118. The security program will generate a command to instruct the SSL server 44 to establish a connection with the database server 76. The connection is made through the internal switch 70 using communications programs 68 and 81. Once a connection is made, the database server 76 will temporarily store the contact information provided. Once the contact information is stored, the connection is terminated. The contact information will be used by the database server 76 to setup the account data 88, account information 89, and other account information for the client 14.

Detail Description Paragraph:
[0119] After the message at block 126 is displayed, the system administrator is contacted to verify the order, at block 128. The system administrator's verification is used as a backup to determine whether the contact information provided by the client 14, such as the domain name, is verifiable through the public registry or other means. If the information is verified by the system administrator, the system administrator will accept the order at block 130, and the account setup process will continue at block 138. If the information is not verifiable, the system administrator will reject the order at block 132. Thereafter, the system administrator accesses the SSL server to generate a message that is transmitted to the client 14 to notify the client 14 the order has been rejected and the reasons for the rejection, at block 134. The reasons can range from the domain name is not found or the information provided was not verifiable. Once the message notifying the client 14 that the order is rejected, the account information that was transferred by the SSL server 44 to the temporary database or processing queue of the database server 76 is deleted, at block 136.

Detail Description Paragraph:
[0122] After the information is verified, the communications program of the SSL server generates a command to transfer the contact information to the database server 76, at block 142. The database server 76 will store all of the contact information to create an account, such as account settings 66, account information 89, authentication information file 90, billing information file 92 and the appropriate task information file 107. The database server 76 will initiate the assignment program 98 to assign a customer ID and password that will be used by the client 14 for accessing services from the system 10, at block 146. Thereafter, the database server 76 assigns the client to one or more servers of group 40 at block 148, depending upon the type of services desired by the client 14. For example, the database server 76 may assign the client 14 to a mail server, a log server, an FTP server, a web server, and SSL server, a real server, a shell server and the like. Once the database server 76 has assigned the services requested by the client 14 to a particular server, the database server 76 will store the assignments in the designated assignment file 102. Thereafter, the database server 76 will generate a configuration file associated with each service that is requested by the client 14. The configuration file will contain the sequences and commands that will be used by the servers of group 40 to configure the predetermined system files 58, daemon 56 and configuration files of the operating system 54 to make the services available to the client 14 as desired. In addition, the database server 76 will create tasks 106 associated with each configuration. The tasks 106 will comprise a set of instructions, commands and sequences that are to be automatically executed by the server to obtain the configuration file and to modify any supporting operating system located on the server to setup and run the application software to deliver the services desired. The tasks 106 are stored in a task file 107 and mapped to the particular server that will be used to provide the service to the client 14.

Detail Description Paragraph:
[0125] After the connection has been established, the SSL server generates the instruction key 74. The key 74 will have a set of commands or instructions that will direct the receiver (i.e., the server receiving the call from the SSL server) to contact the database server 76 to

determine what tasks have to be performed to setup the services on the server to the client 14. If the key 74 is rejected, such as block 158, the process of setting up the account services for the client 14 on that particular server is terminated at block 160. A key may be rejected if the server has reached its capacity, is not working or is in use. Preferably, the SSL server will receive state information from the database server 76 to determine the status of the server before an attempt is made to use the key 74.

Previous Doc Next Doc Go to Doc#